AMENDMENTS TO THE CLAIMS

The claims in this listing will replace all prior versions, and listings, of claims in the application.

1. (Currently amended) A method for migrating a computing process from a first host to a second host, wherein said process discards execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ specific to the first host, and wherein said process receives execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ specific to said second host.

2. (Currently amended) A method as claimed in claim 1 wherein said process discards execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ specific to said first host prior to migration to said second host.

3. (Currently amended) A method as claimed in claim 2 wherein prior to migration a construct is formed comprising application specific execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ of said process.

4. (Currently amended) A method as claimed in claim 3 wherein said construct is formed by a construct operation that suspends all active threads of said process and records application specific execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ of said process.

5. (Original) A method as claimed in claim 4 wherein said construct comprises only data, program code and execution states falling within lists that are passed to said construct operation.

6. (Previously Presented) A method as claimed in claim 3 wherein said construct is provided with an authorizing signature.

7. (Previously Presented) A method as claimed in claim 3 wherein said construct is sent directly to said second host on a communication medium.

8. (Previously Presented) A method as claimed in claim 3 wherein said construct is sent to an intermediary memory storage means, and subsequently to said second host.

9. (Previously Presented) A method as claimed in claim 7 wherein a second process is run on said second host that comprises the data, program code and execution states in said construct.

10. (Currently amended) A method as claimed in claim 9 wherein in said second host a third process is created containing system specific execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ relating to said second host, and wherein said second process assimilates said third process.

11. (Currently amended) A method as claimed in claim 1 wherein said process discards execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ specific to said first host after migration to said second host, but before receiving execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ specific to said second host.

12. (Currently amended) A method as claimed in claim 11 wherein prior to migration a construct is formed comprising execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ of said process.

13. (Currently amended) A method as claimed in claim 12 wherein said construct is formed by a construct operation that suspends all active threads of said

process and records execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ of said process.

14. (Previously Presented) A method as claimed in claim 12 wherein said construct is provided with an authorizing signature.

15. (Previously Presented) A method as claimed in claim 12 wherein said construct is sent directly to said second host on a communication medium.

16. (Previously Presented) A method as claimed in claim 12 wherein said construct is sent to an intermediary memory storage means, and subsequently to said second host.

17. (Previously Presented) A method as claimed in claim 15 wherein a second process is created on said second host that comprises the data, program code and execution states in said construct.

18. (Currently amended) A method as claimed in claim 17 wherein said second process performs a mutate operation that suspends all active threads of said second process and discards system specific execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ relating to said first host.

19. (Original) A method as claimed in claim 18 wherein said second process comprises only data, program code and execution states falling within lists that are passed to said mutate operation.

20. (Currently amended) A method as claimed in claim 19 wherein in said second host a third process is created containing system specific execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ relating to said second host, and wherein said second process assimilates said third process.

21. (Currently amended) A method as claimed in claim 1 wherein said process discards execution states and at least one of data[[,]] and~~/or~~ program code ~~and/or execution states~~ specific to said first host after migration to said second host, and after receiving execution states and at least one of data[[,]] and~~/or~~ program code ~~and/or execution states~~ specific to said second host.

22. (Currently amended) A method as claimed in claim 21 wherein prior to migration a construct is formed comprising execution states and at least one of data[[,]] and~~/or~~ program code ~~and/or execution states~~ of said process.

23. (Currently amended) A method as claimed in claim 22 wherein said construct is formed by a construct operation that suspends all active threads of said process and records execution states and at least one of data[[,]] and~~/or~~ program code ~~and/or execution states~~ of said process.

24. (Previously Presented) A method as claimed in claim 22 wherein said construct is provided with an authorizing signature.

25. (Previously Presented) A method as claimed in claim 22 wherein said construct is sent directly to said second host on a communication medium.

26. (Previously Presented) A method as claimed in claim 22 wherein said construct is sent to an intermediary memory storage means, and subsequently to said second host.

27. (Previously Presented) A method as claimed in claim 25 wherein a second process is created on said second host that comprises the data, program code and execution states in said construct.

28. (Currently amended) A method as claimed in claim 27 wherein in said second host a third process is created containing system specific execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ relating to said second host, and wherein said second process assimilates said third process.

29. (Currently amended) A method as claimed in claim 28 wherein said second process performs a mutate operation that suspends all active threads of said second process and discards system specific execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ states relating to said first host.

30. (Original) A method as claimed in claim 29 wherein said second process comprises only data, program code and execution states falling within lists that are passed to said mutate operation.

31. (Currently amended) A method as claimed in claim 1 wherein the execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ discarded by said process relate(s) to library modules and/or input/output device drivers of said first host.

32. (Currently amended) A method as claimed in claim 1 wherein the execution states and at least one of data[[,]] and/~~or~~ program code ~~and/or execution states~~ received by said process relate(s) to library modules and/or input/output device drivers of said second host.